RESEARCH PAPER

From a Dynamic Image Annotation Process within the Humanities to a Canonical Workflow

Andreas Pfeil[†], Thomas Jejkal[†], Danah Tonne & Germaine Götzelmann

Karlsruhe Institute of Technology, 76344 Eggenstein-Leopoldshafen, Germany

Keywords: FAIR; CWFR; Workflow; Humanities; Automation

Citation: Pfeil, A., et al.: From a dynamic image annotation process within the humanities to a canonical workflow. Data Intelligence 4(2), 386-397 (2022). doi: 10.1162/dint_a_00138

Received: September 18, 2021; Revised: January 13, 2022; Accepted: February 5, 2022

ABSTRACT

One idea of the Canonical Workflow Framework for Research (CWFR) is to improve the reusability and automation in research. In this paper, we aim to deliver a concrete view on the application of CWFRs to a use case of the arts and humanities to enrich further discussions on the practical realization of canonical workflows and the benefits that come with it. This use case involves context dependent data transformation and feature extraction, ingests into multiple repositories as well as a "human-in-the-loop" workflow step, which introduces a certain complexity into the mapping to a canonical workflow.

1. INTRODUCTION

Over the last decade, research data management (RDM) found its way into many research fields and is widely considered as a crucial part of research. Especially the arts and humanities have extensive experience, as they have been doing RDM for a long time in the analogous form of libraries. Basic principles like making research artifacts available in research data repositories, assigning persistent identifiers to them and putting curation processes in place in order to ensure their long term availability are well understood and accepted by scientists from these domains. However, in terms of reproducibility and automation of recurring workflows, the infrastructures in use still have a remarkable gap left necessitating a lot of manual work which is error-prone and leads in most cases to incomplete results. The Canonical Workflow Framework for Research (CWFR) [1] aims for a solution to fill this gap by identifying recurring tasks, making them publicly available and identifiable and linking them.

[†] Corresponding authors: Andreas Pfeil (Email: andreas.pfeil@kit.edu; ORCID: 0000-0001-6575-1022); Thomas Jejkal (Email: thomas.jejkal@kit.edu; ORCID: 0000-0003-2804-688X).

In this context, we are introducing a scientific workflow from the arts and humanities and analyze how it can be represented as a CWFR-compliant workflow. The main goal of this workflow is to ingest digitized manuscripts from the medieval age and annotations that refer to the digital manuscript pages and add heterogeneous information to certain page areas. In doing so, this workflow brings together manual annotations of researchers, as well as annotations representing results from automated analysis methods. The specialized repositories make the data web-resolvable, collaboratively editable and extensible. Multiple pre-processing steps and a feature extraction process are executed before the ingest, generating additional annotations.

Annotations are being used to add translations of sentences or single words, to classify drawings or images and to label layout elements like text blocks. Page XML [2] is being used to describe document image page content in the scope of layout analysis and Optical Character Recognition (OCR) workflows. The Web Annotation Data Model (WADM) is a generic way to store annotations of online resources. Portable Document Format (PDF) also supports annotations using the XML Forms Data Format (XFDF) [3] The WADM has found its way into several domain specific practices used in Mirador [4] and Pelagios [5]. By choosing implementations that support widely supported standards facilitates reuse for other use cases.

After data has been ingested, scientists manually modify and add annotations in the process of their research. In its current state, the workflow is an undocumented series of actions of scientists and data stewards involving different services, user interface applications and command line tools.

This paper describes required steps in order to transform from the currently applied workflow into a CWFR-compliant workflow. It maps contained steps to canonical workflow elements defining their inputs, outputs and extension points. Furthermore, we will elaborate how an integration into the framework, including the capturing of state information, could be achieved and where we see gaps related to our specific use case. We aim to deliver a concrete view on this use case to enrich further discussions on the practical and consistent realization of canonical workflows.

2. DESCRIPTION OF AN ANNOTATION WORKFLOW IN THE ARTS AND HUMANITIES

The workflow includes several phases: pre-processing of a collection of images and metadata, ingest of data into specialized repositories, and continuous post-processing of the ingested data by a human-in-the-loop step. As input, a collection of images and their metadata is provided, coming from a manual digitization process of medieval manuscripts. Depending on the process the images can be contained in a simple folder or may be delivered embedded in PDF together with manual annotations. The manuscript metadata is either already provided in Text Encoding Initiative (TEI) Extensible Markup Language (XML) format [6] or is in some cases given in an Excel Sheet XML (XLSX) format without pre-defined schema and needs to be converted to TEI semi-automatically, which may also involve contacting researchers to ask for missing information.

Pre-processing Phase Regions and associated features are extracted in a standardized XML format, where the format and the method of extraction are both depending on the context. For example, in some datasets the SWATI layout analysis [7] is used to extract layout information in Page XML Format [2].

Ingest of Data The manuscripts are ingested into a research data repository [8] following a custom, use-case specific model. This model defines two elements: resources representing a manuscript, containing all metadata of a manuscript as well as references to all elements of the second kind, the manuscript page resources. Each page resource includes the digitized page, typically in a common image format, as well as a JPEG version of the page and a thumbnail representation for the Web. The creation of the derived images is part of the ingest process and is done by the repository itself. In addition, each page resource refers to the manuscript resource it belongs to in order to be able to resolve relationships in both directions. After ingesting data and metadata into the research data repository, the previously generated XML document (containing extracted features and their associated regions for each page) is transformed into the WADM, which represents annotations in a W3C recommendation [9]. This enables the researchers to access all annotations in a common way using the Web Annotation Protocol (WAP, [10]) in further workflow steps. In the WADM, each annotation references a region on one of the previously ingested page resources and one or more associated features. The resulting Resource Description Framework (RDF) document is then registered in a WAP Server[①][②][③], providing a W3C-recommended interface for accessing WADM annotations [11].

Continuous Post-processing by a Human-in-the-loop Step Domain scientists are manually analyzing the data i.e. using SPARQL Protocol And RDF Query Language (SPARQL) queries [12] and validate, update and add new annotations using specialized user interfaces. Steps in this phase are executed repeatedly over long time spans and form the actual challenge in this workflow as they require a high degree of reproducibility if multiple users are working on the same data. The WAP Server itself does not contain any provenance tracking (besides assigning a modification timestamp to annotations and authors to annotations and features) or version control, such that modifications can not be reproduced.

2.1 Potential of the CWFR Adoption

There is no documentation of the process and no automated execution of steps. It is clear that an executable, state of the art workflow definition would already improve the standardization and automation of the workflow. But there are more issues regarding FAIR data management, which the CWFR concept might be able to solve.

As stated before, the WAP Server includes only very limited provenance tracking and no version control. This does not only limit FAIRness and research data management in general, but it also limits the possibilities

[①] https://github.com/kit-data-manager/wap-server

[②] https://github.com/azaroth42/MangoServer

[③] https://github.com/dlcs/elucidate-server to analyze the resulting data set. As a CWFR realization, all executed steps are being documented using a FAIR Digital Object (FAIR DO) called the CWFR Digital Object (CWFR DO). As this state "captures all relevant information aggregated throughout all states" and "references to information that has been generated throughout these steps" (as described in [1]), it could serve as a technology independent version control (to some yet undefined degree) and a workflow wide provenance tracker without requiring modification of the currently used systems. A gapless, automatically generated documentation of changes would be highly beneficial for the scientists as they can allow rolling back changes applied under wrong assumptions.

As some of the steps are fully manual and unique (in their execution), it is hard to describe and document them well in modern workflow systems, which have their focus on fully automated steps. The CWFR DO has the potential to describe manual steps and their execution properly and in detail, using references to steps, packages and their attributes.

Realizing the described workflow using modern workflow systems will likely involve different systems running in different research centers, like clients doing upload steps or feature extraction and ingests on servers. As the CWFR DO is a FAIR DO, it is accessible for all sites, and with the required authorization, all involved actors are able to update the state, independent of their location, technology or type.

3. TOWARDS A CWFR-COMPLIANT WORKFLOW

Applying CWFR to the workflow is done in two steps. First, we need a formal description to understand which steps will be documented in the CWFR DO. Afterwards we will go into the details of the state itself.

3.1 Workflow Description

The CWFR concept assumes the workflow activities to be representable by recurring, canonical steps (and packages) to reference those in the CWFR DO. To map activities to steps, both need clear borders with defined inputs and outputs. In this section, we will describe the steps that we extracted from our workflow. The result will look similar to existing modern workflow descriptions and is visualized in Figure 1.

The workflow description starts with a number of files, which are being uploaded by a scientist. We assume at this point, that the scientist is supported by a user interface that creates a validated (or verifiable and machine actionable) file structure that suits the workflow. The output of this step is therefore a certain folder structure, representing a manuscript. After each step, the CWFR DO needs to be updated as described in the position paper. Details about the state are described in Section 3.2.

After this first step, two steps of the kind "data transform" are introduced to transform some of the files, if available, depending on their type. Both steps may run in parallel. One transforms PDFs containing annotated images in XFDF [3] into images and annotations (as WADM [9]). The other step transforms a given Excel spreadsheet (XLSX) file into TEI-XML [6], and is unlikely to be automated, as it also involves contacting scientists for missing information as no consistent structure or schema can be assumed within the spreadsheet.

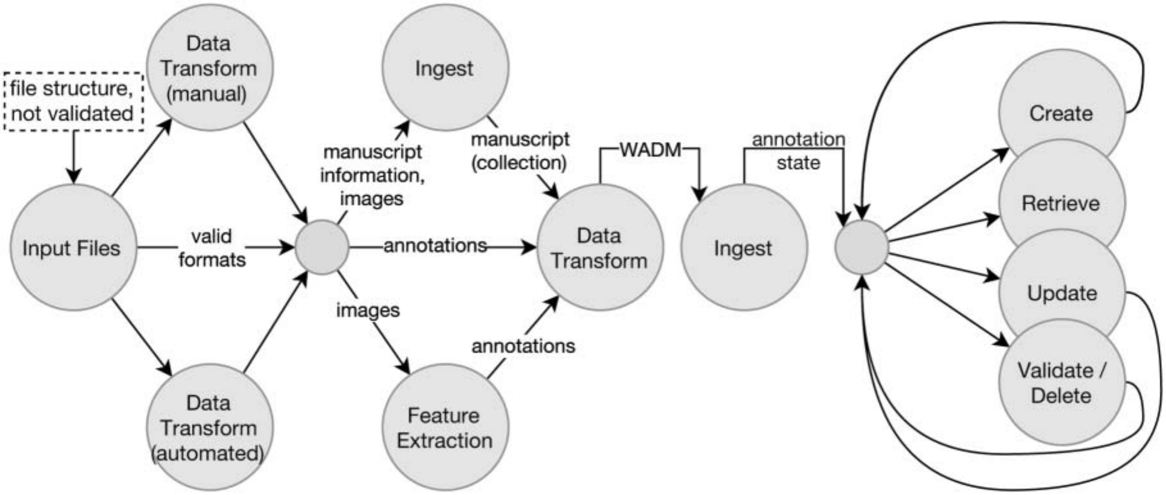

Figure 1. The steps of the workflow as they can be extracted from the current manual steps, connected with arrows defining the package inputs and outputs. To simplify the visual appearance of the illustration, a green dot is used on the left, representing a common state where images, manuscript information and optionally annotations in WADM-Format are available. Similarly, the green dot on the right side represents the slightly changed state of the set of annotations due to the iterations.

The next step is to extract further features from the images and create annotations out of those. The SWATI image analysis [7] is a specialized tool that is driving the step execution in this case. So we expect "swati-feature-extraction" to be a package for the step "feature extraction". The resulting Page XML format will associate annotations with the local file name the annotation belongs to.

The manuscript ingest step is independent to the feature extraction and therefore may run in parallel to it. It takes the TEI metadata and all images as its input and creates a digital representation of the manuscript as a whole inside the repository. After the ingest, TEI metadata is represented as a manuscript resource, which has a reference to all page resources, while each page resource also has a reference to its manuscript resource. The repository also creates web-optimized versions and thumbnails of the images. The result of this package has to be a representation that allows the mapping of the local file names to URLs, as this is needed in the next step. The minimum would therefore be a mapping of filenames to URLs, but it may also be a descriptive collection, similar to the one that exists on a file basis. Except for the actual data input, this step also needs information about how the repository can be accessed (location and authorization). Using the mapping information, it is possible to transform the Page XML files from the feature extraction to the WADM representation (RDF) in such a way that it contains the references to the ingested images (see also figure 1, the step "Data Transform" in the middle).

The WADM is the input for the annotation ingest step. This step is implemented by a "WAP-ingest" package, which obtains a WADM representation and ingest it into a given repository using the WAP. Similar to the manuscript ingest step, it will need access information to the repository as a parameter.

The next step is an abstract human-in-the-loop step: A series of modification steps done by scientists improving the value and accuracy of the annotations using SPARQL and WAP. The input is the type of change and its parameters. We propose the output to be the affected annotation before and after the change within the CWFR DO. To keep the state as small as possible, delta encoding (storing only the differences) could be used. To make versioning possible, it is important that each of those human-in-the-loop steps is documented in the CWFR DO. Therefore, it should be enforced by software.

3.2 CWFR Digital Object

While the workflow defined above contains reusable packages and the important information in inputs and outputs, it does not yet define the CWFR DO and how it is connected to the workflow activities.

According to the position paper, the CWFR DO is a FAIR Digital Object that gets updated after every activity, appending "all relevant information" [1] to the state. This means the object is growing over time and contains the history of executed activities and their inputs and outputs, as illustrated in Figure 2. This allows the reuse of intermediate information within (for example for automated reproduction or re-parameterization) or outside of the workflow context.

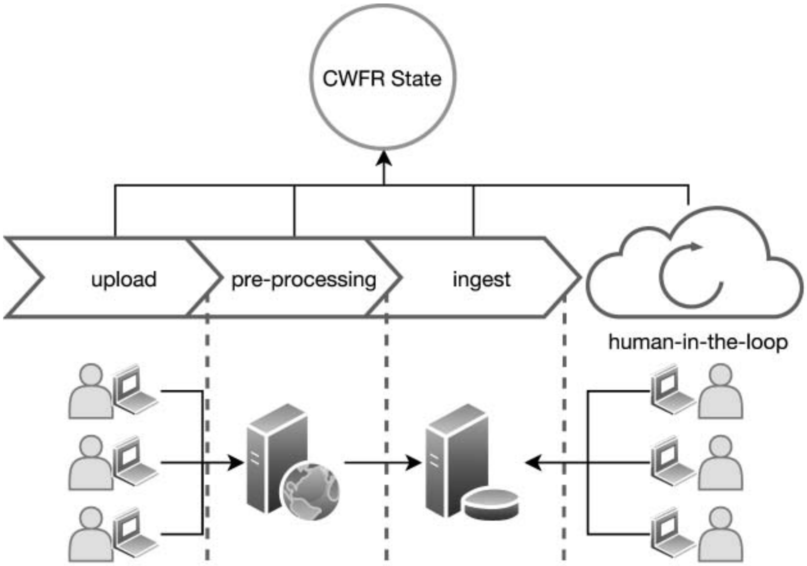

Figure 2. This is a strongly simplified version of the workflow described in this paper to illustrate the idea of the CWFR DO. Each step in a workflow will update the CWFR DO object after execution. This can happen independent of the location and the type of a system, as it is a real FAIR Digital Object. The only requirement is a successful authorization to modify the state.

Every step has inputs, which come from previous steps (with the exception from the first state(s)). A step might also have additional parameters, which are distinct from inputs as they do not originate from previous steps but are constant configuration properties (e.g. algorithm parameters for feature extraction steps or repository location and access information for ingest steps). Outputs of steps might be inputs for other steps, but they do not have to. They are usually reproducible using the inputs and the parameters. The following list elaborates the kind of information the CWFR DO may capture throughout execution in detail.

1. Information produced in a workflow was either generated by a manual step or software (or a combination of both). Such provenance information, including exact software versions, must be stored in detail. Software such as an executable, library or script ideally is described as a independent FAIR digital object which can be "linked to". All common requirements to run the software must be available. A CWFR package, which belongs to one certain step, might already refer to the software executing this package, depending on the concrete design of the CWFR concept. Requirements which are unique to this workflow, like parameters, should be stored in the CWFR DO directly.
2. Each step has parameters (e.g. algorithm parameters for the feature extraction) that define how the inputs will be processed. This information is relevant in order to reproduce the workflow and must be included. The paper already states that the steps are referenced by a PID. Depending on the step, the parameters can usually be added as attributes directly or referenced as a file if this is not possible. This kind of information should be as closely kept to the CWFR DO as possible, as this information is essential.
3. At least the parameters of ingest steps will include location information (e.g. based on the internet protocol), so the step knows where the target repository is located. It consists of relatively few information, like an address and port. This information is not required for reproduction (as any other repository instance can be used). Therefore, it is optional to store it in the CWFR DO. If the repository information already has a PID, it could be used for referencing. On the other hand, as parameters are basically an input to the step anyway, the CWFR should be able to define attributes where this information can be stored.
4. The workflow contains manual (or semi-automated) steps, which are hard to reproduce as a matter of principle. Their results should be documented in the CWFR DO if possible, to maintain at least a certain degree of reproducibility of the following steps. In our workflow, the Data Transform step that transforms XLSX files to TEI files is a semi-automatic step. But in this case the result is stored in the first ingest step in any case. If the CWFR is able to detect this fact (which might not be the case in our workflow because the ingest step is executed afterwards), it might simply store a reference to this repository. Otherwise they should be stored directly within the CWFR DO. The same applies to the manually triggered steps at the end of our workflow. The "Add", "Update" and "Validate/Delete" steps are not reproducible within the annotation store. To use the CWFR as a versioning system and provenance tracker, it must contain this information (for example the author, timestamp, annotation before the change and after the change or a reverse delta or synopsis).
5. There are a lot of text-encoded file formats (TEI, Page XML, etc.) occurring between steps, as well as images. Except for the special cases mentioned in this list, they can be reproduced using the files before the first step and the order of steps and their parameters, if we assume the manual steps to be specially treated as before. From this point of view, this information is not relevant for the CWFR DO.

In case those files are stored anyway as part of the workflow, they might be optionally referenced in the CWFR DO.

6. The origin of an input (the step that produced it) is relevant for the reproduction of a workflow and should be stored in every case. This might be achieved by some kind of reference or a unique, computational identifier (e.g. a hash, which can be computed by several steps with the same result).
7. The first input artifacts in a workflow, which could be referred to as "raw data", have to be stored and documented in the CWFR DO, as they are the key for the reproduction for the workflow.

Only keeping information that can not be automatically reproduced, minimizes the size of the CWFR DO. It is still a matter of discussion how exactly the CWFR DO should be structured, although the position paper makes an abstract proposal. As the CWFR DO is a FAIR Digital Object, there are multiple ways to store information:

The PID record Each FAIR Digital Object has a persistent identifier (PID), which is registered at a PID service. Such a PID service stores a record for each PID. As long as the PID exists (meaning as long as it is resolvable), there is also a record for this PID. This record is usually being considered a key-value map (or table) and can store limited amount of simple information. Due to the limitation of size (in favor of fast PID resolving) and the record structure, it will not be the ideal place for the information listed before. The information in this record should instead describe at least (a) that this PID is pointing to a CWFR DO, (b) whether the workflow is still updating the actual object this record references to, so machines know they have to fetch updates later on and (c) relations to other CWFR DOs or other FAIR DOs, so a PID graph can easily be built up even by clients which do not understand CWFR DO objects.

The CWFR Digital Object The PID record references the actual object, which is of a format yet to be defined and should contain the information above (keeping it "close" to the state) or references to it. The latter might be needed or useful in case the information is stored elsewhere anyway or requires specialized repositories. It can be expected that the CWFR DO will not only need to store textual information and references, but actual data. Concepts like RO-Crate allow bundling arbitrary data, contextual information and references as well and should be considered candidates or inspiration for the CWFR DO structure [13, 14]. The RO-Crate specification explicitly defines how workflows can be represented using the format[④].

4. DISCUSSION AND CONCLUSION

We have demonstrated the feasibility to apply the CWFR concept to a nonlinear, practical workflow with manual steps. Analyzing a practical workflow like this will help to define a CWFR DO layout that will be able to store required information in real-world use cases, as it allows in-depth analysis of possible pitfalls and identification of parts perceived as unique to a workflow. We identified recurring steps and represented

[④] RO-Crate Specification 1.1—Workflows: https://www.researchobject.org/ro-crate/1.1/workflows.html the current workflow as a workflow graph. Then we described the inputs and outputs of those steps and which information should be stored inside the CWFR DO to maintain reproducibility. Analyzing this use case, open questions were identified regarding the steps and packages definitions as well as the inner workings of the CWFR DO.

Even though the workflow was not formally defined in beforehand, identifying recurring, generic steps was possible and a similar process to creating a formal workflow definition. It was unclear though what conditions a step will define for a package. For example, if a step defines the exact inputs and outputs for each of its packages, custom formats or configurations will not be possible within its packages. This would affect for example the Data Transform step, which we assumed to be generic over input data and target formats, as well as possibly different configuration parameters. In our model, the package is then a specialized entity, implementing one certain transformation type. It is needed to properly define this principle to allow consistent realizations.

When analyzing the information to be stored within the CWFR DO, it was unclear how the relation between steps can be stored. Steps have PIDs which are stored in the CWFR DO after the according step has been executed. It was already described which attributes should be stored in the CWFR DO together with those identifiers. But it is unclear there an input attribute (e.g. a file) originated from. Adding such information would allow to reconstruct the full execution graph. It is also required for the versioning system and provenance tracking that was described.

To support the CWFR concept from a technical point of view, the existing clients and server applications would have to be modified (or encapsulated with new software) to enable each software to update of the CWFR DO. Supporting libraries and general functionalities can help to simplify the implementation tasks. If the software received those changes, the handling would not necessarily be different from a scientist point of view. But it can offer new possibilities to researchers to understand how data has changed, and to use the addressed, previously missing functionalities like the rollback of changes and better provenance tracking.

ACKNOWLEDGEMENTS

This research has been supported by the research program “Engineering Digital Futures” of the Helmholtz Association of German Research Centers and the Helmholtz Metadata Collaboration Platform (HMC). We thank all contributors from the Research Data Alliance, the FAIR DO Forum, and HMC as well as potential users for fruitful discussions and feedback. The humanities use case originates from the Collaborative Research Center 980 “Episteme in Motion”[⑤], and thus we thank all members for the valuable insight into their daily research work.

[⑤] https://www.sfb-episteme.de/

AUTHOR CONTRIBUTIONS

D. Tonne (danah.tonne@kit.edu), G. Götzelmann (germaine.goetzelmann@kit.edu) and T. Jejkal (thomas.jejkal@kit.edu) wrote the introduction and the workflow description. A. Pfeil (andreas.pfeil@kit.edu) and T. Jejkal created and wrote about the adoption of CWFR, with contributions from all authors, and wrote the adoption and discussion sections.

REFERENCES

[1] Hardisty, A., Wittenburg, P. (eds.): Canonical Workflow Framework for Research (CWFR)—position paper—version 2, December 2020. Working paper. Available at: https://osf.io/9e3vc/. Accessed 12 March 2021

[2] Pletschacher, S., Antonacopoulos, A.: The PAGE (page analysis and ground-truth elements) format framework. In: The 20th International Conference on Pattern Recognition, pp. 257–260 (2010)

[3] International Organization for Standardization (ISO): ISO 19444-1:2016(en), Document management—XML forms data format—Part 1: Use of ISO 32000-2 (XFDF 3.0). Available at: https://www.iso.org/obp/ui/#iso:std:iso:19444:-1:ed-1:v1:en. Accessed 4 January 2022

[4] van Zundert, J.: On not writing a review about Mirador: Mirador, IIIF, and the epistemological gains of distributed digital scholarly resources. Digital Medievalist 11(1), 5 (2018)

[5] Simon, R., et al.: Linked data annotation without the pointy brackets: Introducing Recogito 2. Journal of Map & Geography Libraries 13(1), 111–132 (2017)

[6] TEI Consortium: TEI: Text Encoding Initiative. Available at: https://tei-c.org/. Accessed 10 September 2021

[7] Chandna, S., et al.: Software workflow for the automatic tagging of medieval manuscript images (SWATI). In: Proceedings of SPIE 9402, Document Recognition and Retrieval XXII, Article No. 940206 (2015)

[8] Jejkal, T., Hartmann, V.: KIT data manager—Base repository service. Available at: https://github.com/kit-data-manager/base-repo. Accessed 4 January 2022

[9] Sanderson, R.: Web annotation protocol. Available at: https://www.w3.org/TR/annotationprotocol/. Accessed 10 September 2021

[10] Sanderson, R., Ciccarese, P., Young, B.: Web annotation data model. Available at: https://www.w3.org/TR/annotation-model/. Accessed 10 September 2021

[11] Tonne, D., et al.: Ein Web annotation protocol server zur Untersuchung vormoderner Wissensbestände. In: DHd 2019 Digital Humanities: Multimedial & Multimodal. Available at: https://zenodo.org/record/4622129. Accessed 4 January 2022

[12] The W3C SPARQL Working Group: SPARQL 1.1 overview. Available at: https://www.w3.org/TR/sparql11-overview/. Accessed 4 January 2022

[13] Barratt, J., Rono, S., Walsh, P.: Frictionless data and data packages. Available at: https://zenodo.org/record/1301152. Accessed 26 July 2022

[14] Goble, C., Sefton, P., Soiland-Reyes, S.: Research object crate. Available at: https://www.researchobject.org/. Accessed 16 September 2021

AUTHOR BIOGRAPHY

Andreas Pfeil received his Master's degree in Computer Science at Karlsruhe Institute of Technology (KIT), Germany. His Master's thesis dealt with distance metrics for and the visualization of unexplored, unstructured metadata in the context of historical manuscripts. Now he is working at KIT within the Helmholtz Metadata Collaboration (HMC) platform on the realization of FAIR Digital Objects in the Helmholtz Association.
ORCID: 0000-0001-6575-1022

Thomas Jejkal studied Computer Science at Baden-Württemberg Cooperative State University, Germany. After his studies he was working on the topic of Grid Computing within the framework of the German Grid Initiative D-Grid, mainly focusing on distributed computing and Open Grid Service Architecture Data Access and Integration (OGSA-DAI). From the year 2010 he started working on cross-disciplinary software solutions in the field of research data management. Currently, he coordinates the working package "FAIR Data Commons" of the Helmholtz Metadata Collaboration (HMC) Platform responsible for defining and implementing base services and generic processes to harmonize metadata management in the Helmholtz Association. He is also active in the Research Data Alliance, where he was co-chair of the Research Data Repository Interoperability Working Group until 2017.
ORCID: 0000-0003-2804-688X

Danah Tonne is a Deputy Department head at the Steinbuch Centre for Computing (SCC) at Karlsruhe Institute of Technology (KIT), Germany, and holds a Ph.D. in Computer Science. Her thesis focused on undetected faults in bit preservation architectures for long-term data storage. As a project lead within the Collaborative Research Centre 980 "Episteme in Motion", she is responsible for the development of a sustainable research data repository including an annotation infrastructure for enrichment, analysis, and visualization.
Photo: Christina Stivali
ORCID: 0000-0001-6296-7282

Germaine Götzelmann is a Research Associate at the Steinbuch Centre for Computing (SCC) at Karlsruhe Institute of Technology (KIT), Germany, and is working towards a doctoral degree in the field of computational philology. Her research interests lie within the topics of sustainable research data management and big data analysis in various research areas of digital humanities. Photo: Christina Stivali
ORCID: 0000-0003-3974-3728